Distributed Data Management Support for Collaborative Computing

S.P. Olesen

S.E. Chodrow

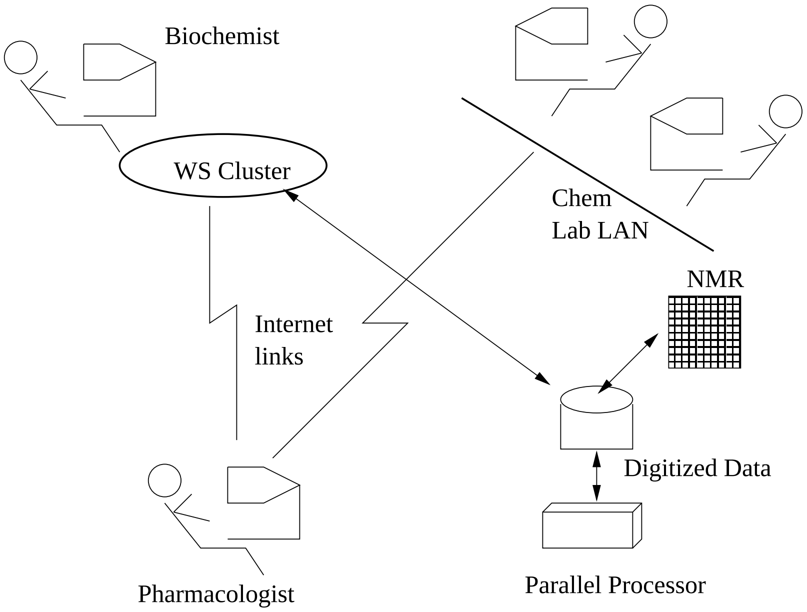
M. Grigni

V.S. Sunderam

Department of Mathematics and Computer Science
Emory University, Atlanta, GA, USA

Research supported in part by NSF grant ASC-9527186

Overview



- Displays
- AV conferencing
- Compute space
- Data management support
- Communication system

Application Scenario

Use shared data for computation

⇓

View output

⇓

Report results in electronic notebook

⇓

Collaborative edit journal paper

Collaborative Computing Data Space

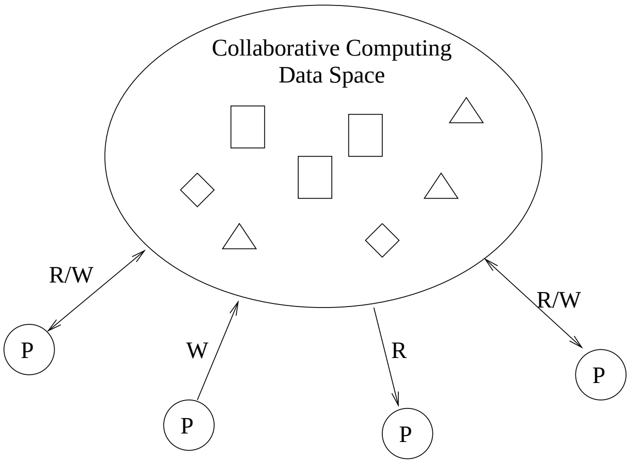
Objectives

- Shared data space for collaborative environments
- Interface to shared objects
- Concurrent access
- User-level tools
- Application Programming Interface

CCDS - Requirements

- Consistent
- Distributed
- Heterogeneous
- Fault tolerance
- Performance, scalability
- Security

CCDS - Model



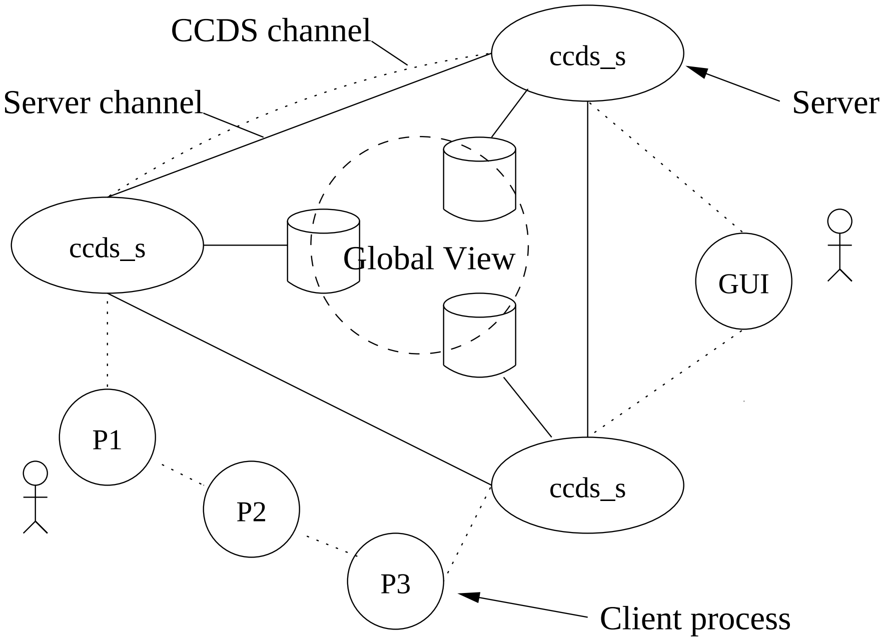
Shared Virtual Objects

- Native file
- Replicated file
- Directory
- Active, Device
- Pipe, FIFO

Virtual Object Operations

- Import / un-import
- Create new object
- Object attributes (name, type, owner, permissions, token, coherence, data)
  - Coherence (real-time, on-demand, interval, never)
- Application programming interface
- Standard file operations
- Object specific operations

CCDS - System Architecture



CCDS - Design

- No root access required
- One server per machine per session
- Native file system
- Group and point-to-point communication
- Flat global name space
- Distributed name service

CCDS - API

- `int ccds_init( argc, argv )`
- `void ccds_leave( )`
- `int ccds_open( name, ...  )`
- `void ccds_close( fdes )`
- `void ccds_delete( name )`
- `int ccds_read( fdes, buf, nbyte )`
- `int ccds_write( fdes, buf, nbyte )`
- `long ccds_lseek( fdes, offset )`
- `int ccds_lock( fdes, size )`
- `int ccds_olock( fdes, offset, size )`
- `int ccds_unlock( fdes )`
- `int ccds_import( srcname, dstname )`
- `int ccds_unimport( srcname, dstname )`
- `int ccds_property( name, buf )`
- `int ccds_ls( buf, nitem )`
- `void ccds_notify( cb )`

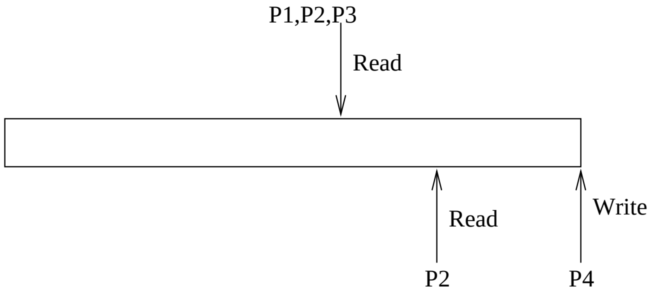
CCDS - Pointers

`int ccds_open(name, oflag, mode, view, lflag)`

File handle can not be shared

`view` is Shared or Local

- Shared: Processes share read or write file pointer.
- Local: Only process can access file pointer.



Summary and Future

- Proposed framework for CCDS
- Architecture
- Implementation
- Integration with collaborative computing systems